

---

**<SJTU>**

---

**<在线文档工具>**  
**软件架构文档**

版本 **<1.0>**

<在线文档工具>	Version: <1.0>
软件架构文档	Date: <15/06/2022>

修订历史记录

日期	版本	说明	作者
<15/06/2022>	<1.0>	<初步完成软件架构文档>	<刘骏霖，李思旷，张子谦，姜凯>

<在线文档工具>	Version: <1.0>
软件架构文档	Date: <15/06/2022>

## 目录

1. 简介	4
1.1 目的	4
1.2 参考资料	4
2. 用例视图	5
3. 逻辑视图	6
3.1 概述	6
3.2 在构架方面具有重要意义的设计包	7
3.2.1 Business Service	7
3.2.2 Middleware	7
4. 部署视图	8
5. 实现视图	8
6. 技术视图	8
7. 质量属性的设计	9

<在线文档工具>	Version: <1.0>
软件架构文档	Date: <15/06/2022>

# 软件架构文档

## 1. 简介

### 1.1 目的

本文档用于阐述在线编辑工具的软件架构设计，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。

### 1.2 参考资料

该软件架构文档参考了前期开发过程中，开发团队所撰写的所有文档，包括但不限于：

- 《在线文档工具 - Vision文档》
- 《在线文档工具 - 用例模型》
- 《在线文档工具 - 软件需求规约》

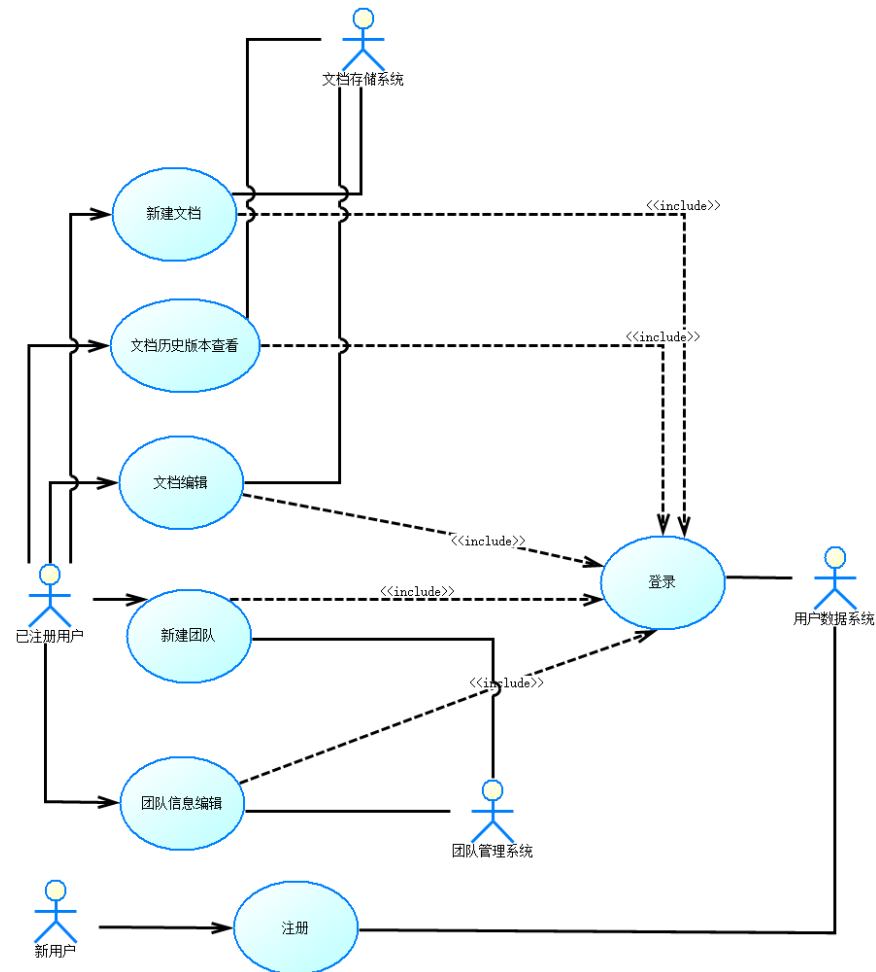
上述文档均在附在“FinalRelease”目录下的文档目录中。为撰写这份文档提供帮助的课程资料有：

- 《04\_面向对象分析.pdf》
- 《05\_设计工程.pdf》
- 《06\_面向对象设计.pdf》
- 《07\_结构化分析与设计.pdf》

这些资料均在课程文件中可以获取到。

<在线文档工具>	Version: <1.0>
软件架构文档	Date: <15/06/2022>

## 2. 用例视图



产品的用例视图中涉及两类用户：

- 新用户，指新接触产品的，未拥有产品账号信息的用户；
- 已注册用户，指拥有本产品账号，并能使用产品提供的全部功能的用户。

用例视图中涉及的产品子系统有：

- 用户数据系统，用于存储用户账号信息，并在登录、注册等场合中提供账号验证服务；
- 文档存储系统，用于存储用户所有的文档内容，包括文字信息，以及插入的图片、视频等多媒体文件；
- 团队管理系统，用于存储团队的信息，支持新增团队的信息存储、修改团队信息时的信息更新与修改操作鉴权。

用例可依据涉及的功能需求分为两大类：

- 与在线文档功能相关的用例
  - 新建文档：用户创建一份新的文档并保存至自己的用户数据中；
  - 文档历史版本查看：用户查看已有文档的历史版本信息，并可以根据需要将文档回滚至历史版本；
  - 文档编辑：支持用户修改、删除已有文档，其中修改文档包括但不限于修改已有内容，在电子表格中插入普通文本、数字、日期、货币等信息，在文档项目中插入文字、图片、表格等内容。
- 与团队管理功能相关的用例

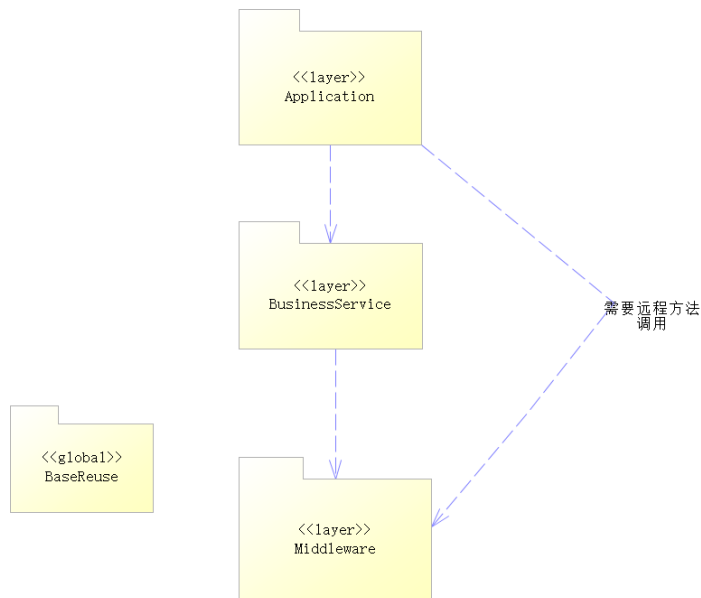
<在线文档工具>	Version: <1.0>
软件架构文档	Date: <15/06/2022>

- 新建团队：支持用户创建新团队并命名；
- 团队信息编辑：支持用户向团队中添加团队成员、添加共享文档等。

### 3. 逻辑视图

#### 3.1 概述

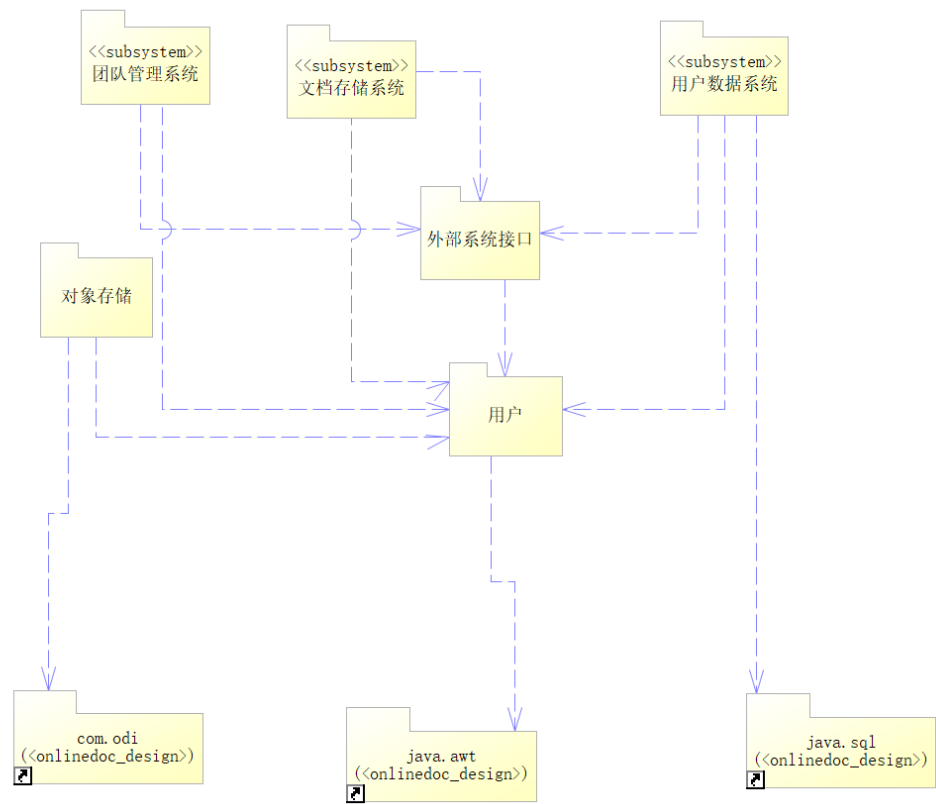
产品的架构逻辑为，Application 依赖于 Business Service 依赖于 Middleware，Application 同时依赖于 Middleware 的远程调用。产品选择了分层架构模式。



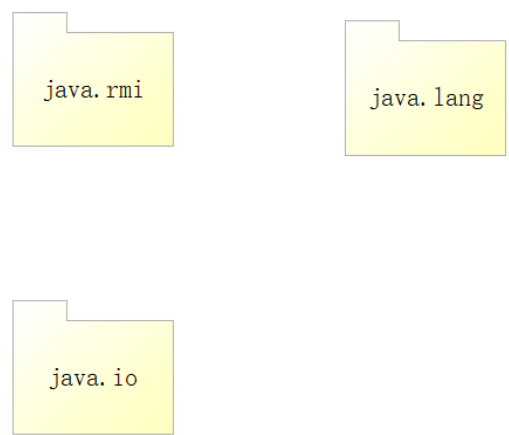
<在线文档工具>	Version: <1.0>
软件架构文档	Date: <15/06/2022>

3.2 在构架方面具有重要意义的设计包

3.2.1 Business Service

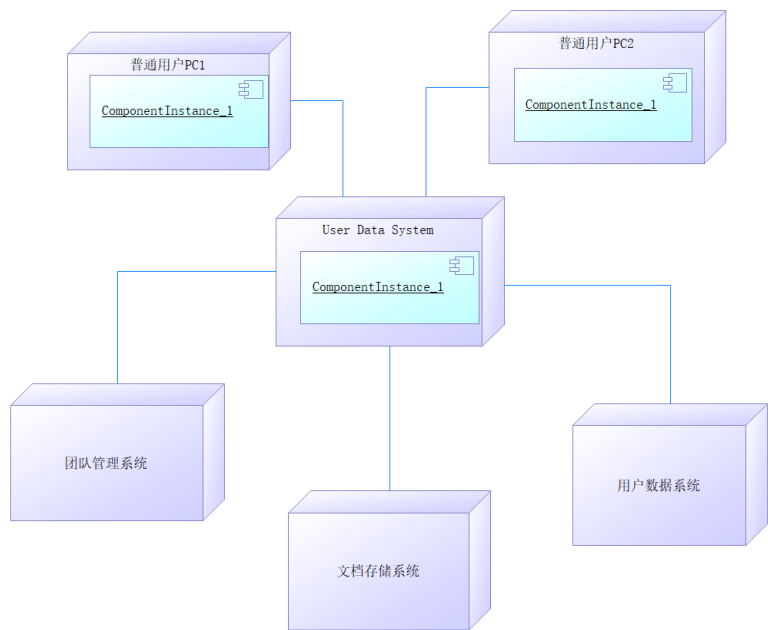


3.2.2 Middleware



<在线文档工具>	Version: <1.0>
软件架构文档	Date: <15/06/2022>

4. 部署视图



用户数据服务与普通用户设备连接，接受用户请求，同时与用户数据系统、文档存储系统、团队管理系统连接，以为用户提供服务。

5. 实现视图



用户直接与服务器相连，向服务端发送请求并由服务端提供相应服务。

6. 技术视图

- 开发过程中的技术视图可由如下列表给出
- 编程语言: JavaScript, Java, PHP
  - 开发工具: JetBrains 系列 IDE
  - 数据库: SQLite



<在线文档工具>	Version: <1.0>
软件架构文档	Date: <15/06/2022>

## 7. 质量属性的设计

软件架构明确了整个系统的结构组成与详细分工，其用例视图明确了系统面向用户的核心功能，使得研发过程中对功能的开发更具有针对性；逻辑视图的设计将系统分解为从问题域中得到的一系列关键抽象，即对象、类和包，这些抽象为系统提供了功能分析，在本次设计中采用分层架构模式进行设计，将不同层次的功能部分抽象出来并分别加以详细说明，从而使得整个架构更加清晰，更具有可扩展性；部署视图的设计则从物理层面明确了本次开发中系统的详细结构，保证在将软件系统部署到不同环境时不必大量修改源代码，提高了可移植性。